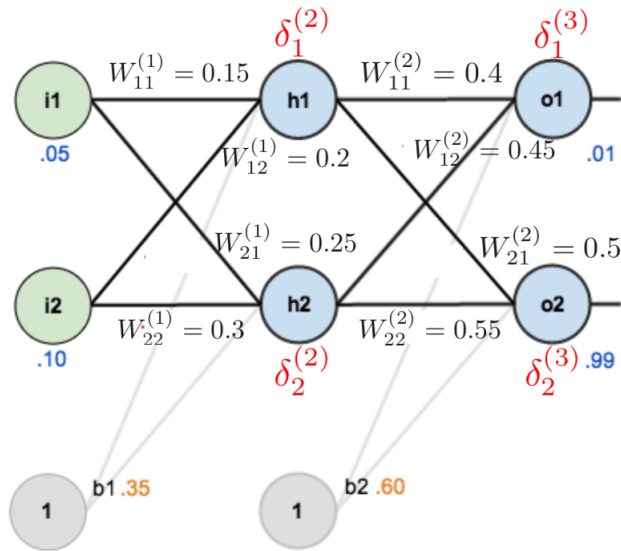


Practice Problem

Perform one iteration of forward propagation and backpropagation in the following neural network, assuming learning rate $\alpha = 0.5$. We also assume a sigmoid activation function $g(x)$ for the hidden nodes, where $g'(x) = g(x)[1 - g(x)]$.

**Backpropagation Implementation**

- For each node i in output layer L : $\delta_i^{(L)} = (\alpha_i^{(L)} - y_n) f'(z_i^{(L)})$
- For each (hidden) node i in layer $l = L - 1, L - 2, \dots, 2$: $\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) f'(z_i^{(l)})$
- Compute the desired partial derivatives as: $\frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial W_{ij}^{(l)}} = \alpha_j^{(l)} \delta_i^{(l+1)}$, $\frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial b_i^{(l)}} = \delta_i^{(l+1)}$
- Update the weights as: $W_{ij}^{(l)} := W_{ij}^{(l)} - \alpha \frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial W_{ij}^{(l)}}$, $b_i^{(l)} := b_i^{(l)} - \alpha \frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial b_i^{(l)}}$

